

一种具有查错功能的 10B/8B 解码器设计

邹 陈¹, 黄 鲁¹, 张步青²

(1 中国科学技术大学 电子科学与技术系, 安徽 合肥 230027; 2 中国科学院 自动化研究所, 北京 100190)

摘 要: 研究了 8B/10B 的原理及其特点, 分析编码的内在关联性和逻辑性, 给出了一种具有查错功能的 10B/8B 并行逻辑解码方法, 并且使用 modelsim10.2 进行仿真, 使用 Synopsys 的 synthesis 利用 TSMC130 工艺进行综合。结果表明其查错功能正确, 并且相对于现有 8B/10B 解码方法, 逻辑运算量小, 速度快。

关键词: 10B/8B 解码; 可查错; 逻辑优化

中图分类号: TN492

文献标识码: A

文章编号: 1000-7180(2016)09-0028-04

A New Kind of 8B/10B Decoding Design Having Error Checking

ZOU Chen¹, HUANG Lu¹, ZHANG Bu-qing²

(1 Department of Electronic Science and Technology, University of Science and Technology of China, Hefei 230027, China; 2 Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Based on the theory of the 8B/10B decoding, this paper makes a deep research on logic analysis, inherent relationship of the decoder. It was implemented a new method of the logical optimal design of an 8B/10B decode having error checking. And the simulation results show that it can reduce the area of the circuit, improve the efficiency of encode and work well when check error.

Key words: 10B/8B decode; error checking; optimal design

1 引言

为了解决高速数据传输过程中的数据失真, 传输错误等问题, 8B/10B 编码方式被广泛应用于各类高速串行数据传输系统中。该编码由 IBM 公司的 Albert X. Windmer 和 Peter A. Franzsenk 最先提出^[1]。这种编码方式转换密度高, 便于时钟恢复, 易于快速同步对齐, 可以保持电路中的 DC 平衡, 提高传输的正确率^[2-3]。传统 8B/10B 编解码方案采用查找表法^[4-5], 将输入的数据按照编码表直接映射成最终编码, 或者是将编码输入后直接映射成原始数据值。该方案易于实现但是消耗逻辑资源太大, 编码效率比较低。本文研究了 8B/10B 的内在原理及其特点, 分析内在关联性和逻辑性, 给出了一种具有查错功能的 8B/10B 并行逻辑解码方法。

2 编码方式原理

8B/10B 编码的输入数据共由 256 种数据码组和 12 种特殊码组组成, 其简写形式分别为 Dx.y(数据码组)和 Kx.y(特殊码组), x 和 y 分别表示 8 bit 数据的低 5 位和高 3 位的十进制值。其编码的五三分组如图 1 所示。

D25.3	HGF	EDCBA
	011	11001
	Y=3	X=25

图 1 8B/10B 编码的五三分组

8B/10B 编码采用采用游程值 RD(Running Disparity)来表示编码后码组的极性, 其中正游程 RD+ 表示编码后码组里 1 的个数大于 0 的个数, 负游程 RD- 表示编码后码组里 0 的个数大于 1 的个数。以此

种方式来保证输出码组的直流平衡(1 与 0 个数相等). 更新 RD 的规则如下:如果编码后码组中 1 的个数多于 0 的个数,或 6B 编码为 6'b000111,或 4B 编码为 4'b0011,则将 RD 值更新为 RD+;如果编码后码组中 1 的个数少于 0 的个数,或 6 bit 编码为 6'b111000,或 4 bit 编码为 4'b1100,则将 RD 值更新为 RD-;否则 RD 值保持不变.

首先进行 5B/6B 编码,每一组 5 bit 对应 RD+ 和 RD-两组 6 bit 码组(abcd ei),根据输入 RD 值将 5 bit 码组编码成 6 bit 码组,按照编码后的 6 bit 码组极性更新 RD 值,然后进行 3B/4B 编码. 每一组 3 bit 码组对应 RD+ 和 RD-两组 4 bit 码组(fghj),根据 5B/6B 编码后更新的 RD 值将 3 bit 码组编码成 4 bit 码组,最后根据 4 bit 极性更新 RD 值,更新的 RD 值作为下一个 8 bit 编码的 RD 输入值. 最终将编码后的 6 bit 码组和 4 bit 码组组成 10 bit 码组(abcd eifghj),8B/10B 编码原理如图 2 所示. 其解码过程则是此过程的逆向.

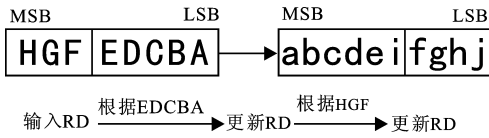


图 2 8B/10B 编码原理

3 8B/10B 解码设计

根据介绍的 8B/10B 编码方法,设计解码模块构造如图 3 所示.

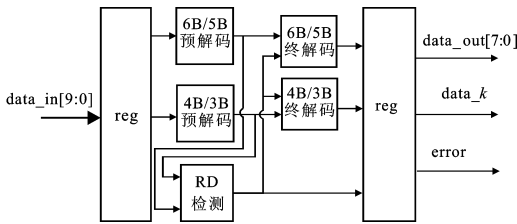


图 3 解码模块构造答题结构图

对于整体的输入信号,首先进行 6B/5B 预解码和 4B/3B 预解码. 不同的输入值会自动的得到相对应的解码值和 RD 值. 进行 10B/8B 解码是不需要进行 RD 值的传递,但是在工作过程中,可以通过检查 RD 的值对错来判断是否传输中出现了错误. 整体输入为 abcdeifghj,输出为 HGFEDCBA.

对于 6B/5B 解码,首先分成 abcd 与 ei 进行处理, p_{22} 表示 abcd 中 1 和 0 的个数相同, p_{13} 表示 abcd 中有 3 个 0 和 1 和 1,而 p_{31} 则为 3 个 1 和 1

个 0. 变化规则如下:

$$A = a \oplus (p_{22} * b_N * c_N * (e = i) + c_N * d_N * e_N * i_N + p_{13} * d * e * i + p_{22} * a_N * c_N * (e = i) + p_{13} * e_N + a_N * b_N * e_N * i_N + p_{31} * i)$$

$$B = b \oplus (p_{22} * b * c * (e = i) + c_N * d_N * e_N * i_N + p_{13} * d * e * i + p_{22} * a * c * (e = i) + p_{12} * e_N + a * b * e * i + p_{31} * i)$$

$$C = c \oplus (p_{22} * b * c * (e = i) + c_N * d_N * e_N * i_N + p_{13} * d * e * i + p_{22} * a_N * c_N * (e = i) + p_{13} * e_N + a * b * e * i + p_{31} * i)$$

$$D = d \oplus (p_{22} * b_N * c_N * (e = i) + c_N * d_N * e_N * i_N + p_{13} * d * e * i + p_{22} * a * c * (e = i) + p_{13} * e_N + a * b * e * i + p_{31} * i)$$

$$E = e \oplus (p_{22} * b_N * c_N * (e = i) + c_N * d_N * e_N * i_N + p_{13} * d * e * i + p_{22} * a_N * c_N * (e = i) + p_{13} * e_N + a_N * b_N * e_N * i_N + p_{31} * i)$$

同时模块中会根据 RD 的计算公式来获取 RD 的值.

$$RD_{6b_neg} = p_{22} * e_N * i_N + p_{13} * e_N + a_N * b_N * c_N * d * e * i$$

$$RD_{6b_pos} = p_{22} * e * i + p_{13} * e + a * b * c * d_N * e_N * i_N$$

模块会将上一组解码后得到的 4B/3B 正确的 RD 值进行保留后输入该模块,根据 RD 的更新法则来选择对应 6b 的 RD 值,如果输入的 RD 值为 0,且 RD_{6b_pos} 值为 0,那么出现 RD_err ,同时传递给 4B/3B 的 RD 值为 0;如果输入的 RD 值为 0,且 RD_{6b_pos} 值为 1,那么正常,传递给 4B/3B 的 RD 值为 1;如果输入的 RD 值为 1,且 RD_{6b_neg} 值为 0,那么出现 RD_err ,同时传递给 4B/3B 的 RD 值为 1;如果输入的 RD 值为 1,且 RD_{6b_neg} 值为 1,那么正常,传递给 4B/3B 的 RD 值为 0.

对于 4B/3B 解码,变化规则如下:

$$F = f \oplus (c_N * d_N * e_N * i_N * (h \neq j) + f * h * j + f_N * g_N * h * j + f * g * j + f_N * g_N * h_N + g * h * j)$$

$$G = g \oplus (c_N * d_N * e_N * i_N * (h \neq j) + f_N * h_N * j_N + f_N * g_N * h * j +$$

$$f * g * j + f_N * g_N * h_N + \\ g_N * h_N * j_N)$$

$$H = h \oplus (c_N * d_N * e_N * i_N * (h \neq j) + \\ f * h * j + f_N * g_N * h * j + \\ f * g * j + f_N * g_N * h_N + \\ g_N * h_N * j_N)$$

同时模块中会根据 RD 的计算公式来获取 RD 的值。

$$RD_{4b_neg} = f_N * g_N * h * j + \\ f_N * h_N * j_N + f_N * g_N * j_N + \\ f_N * g_N * h_N + g_N * h_N * j_N \\ RD_{4b_pos} = f * g * h_N * j_N + \\ f * h * j + f * g * j + \\ f * g * h + g * h * j$$

模块会根据前面解码的 6B/5B 生成 RD 值进行判断,如果输入的 RD 值为 0,且 RD_4b 值为 0,那么出现 RD_err,同时传递给下一组解码的 6B/5B 的 RD 值为 0;如果输入的 RD 值为 0,且 RD_4 b_pos 值为 1,那么正常,传递给下一组解码 6B/5B 的 RD 值为 1;如果输入的 RD 值为 1,且 RD_4 b_neg 值为 0,那么出现 RD_err,同时传递给下一组解码 6B/5B 的 RD 值为 1;如果输入的 RD 值为 1,且 RD_4 b_neg 值为 1,那么正常,传递给下一组解码的 6B/5B 的 RD 值为 0。

解码中也确定了特殊码组. 8B/10B 中定义了 12 种有效的控制码组,分别为 K28. 0、K28. 1、K28. 2、K28. 3、K28. 5、K28. 6、K28. 7、K23. 7、K27. 7、K29. 7、K30. 7. data_k 有效时则表明当前为有效的特殊码组。

$$data_k = c_N * d_N * e_N + i_N + p_{13} * e_N * i * g * h * j + \\ p_{31} * e * i_N * g_N * h_N * j_N + c * d * e * i$$

4 查错功能分析

对于 8B/10B 编码,能够检测出来的错误主要是两种:RD 值不符合实际的要求,或者是出现了单比特位的突变错误. 这两种错误在本设计中都能够进行检测. 对于 RD 的错误,每个 6B/5B 与 4B/3B 模块都会进行 RD 值的计算,然后对于 RD 值会先进行 6B/5B 的 RD 值检测,然后进行 4B/3B 的 RD 值检测:如果输入的 RD 值为 0,且 RD_6b_pos 值为 0,那么出现 RD_err;如果输入的 RD 值为 1,且 RD_6b_neg 值为 0,那么出现 RD_err;根据前面解码的 6B/5B 生成 RD 值进行判断,如果输入的 RD 值为 0,且 RD_4b_pos 值为 0,那么出现 RD_err;根据前

面解码的 6B/5B 生成 RD 值进行判断,如果输入的 RD 值为 1,且 RD_4b_neg 值为 0,那么出现 RD_err. 以上的错误检测都是进行的 RD 值错误检测。

$$rd_err = RD_N * RD_{6b_posN} + \\ RD * RD_{6b_negN} + RD_{6b} * RD_{4b_negN} + \\ RD_{6bN} * RD_{4b_posN}$$

对于单比特位的突变造成的错误,则是适应的 code_err 来进行表示. Code_err 主要是将一些特殊情况进行列出,并且如果在进行 RD 值检测时,发生了两组 RD 极性相同. 主要如下面表达式所显示:

$$code_err = a * b * c * d + RD_{6b_pos} * RD_{4b_pos} + \\ RD_{6b_neg} * RD_{4b_neg} + f * g * h * j + \\ a_N * b_N * c_N * d_N + \\ f_N * g_N * h_N * j_N$$

5 仿真结果分析

采用上述并行逻辑设计的 8B/10B 解码设计方法,对 256 种数据码组和 12 种控制码组分别进行 8B/10B 解码,部分仿真结果如图 4 所示。

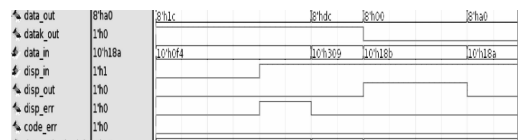


图 4 解码模块仿真图

图 4 中,首先输入数据为 10'h0f4,该数据经过查表可以知道,解码数据值应该是 8'h1c,同时 RD 值为正. 如果输入的上一次解码的 RD 值为负,那么符合 RD 值的更新原理,因此此时没有出现 disp_err 或者是 code_err. 在如果更改了输入的上一次 RD 值,将其改为正后,依然会输出正确的解码,但是会显现 disp_err.

如仿真图 4 所示,解码模块中会输入上次解码后留下来的 RD 的值. 该功能的完成是将模块的输出放入寄存器中进行寄存,在时钟上升沿到来读入数据的时候同时将 RD 的值输入给该模块,用于检查 RD 更新是否出现了错误. 如果输入的 RD 值与当前的 10 位码解出的 RD 值不同,那么则为 RD 更新错误. 如果该 10 位码为特殊字符码,输出 data_k 则为高电平表明为特殊字符码. 如果是简单的数据码,输出 data_k 则为低电平表明为普通的数据码. 如果出现了 10 位数据中一位突变不符合 8b/10b 编码规则的,则会 code_err 变为高电平有效状态预示数据中出现了错误^[6-7].

仿真分析表明,解码结果与码表一致,同时能够检查 RD 值或者是单比特位的错误.表 1 给出了与同类解码电路设计的逻辑资源占用和最高工作频率比较(下面的比较数据使用到了 TSMC130 工艺下的标准单元,由于该模块收外部的触发器控制,主要频率受总体的限制),从中可以看出,本文的并行逻辑优化解码方法使用的逻辑资源为 30LE(Logic Elements),虽然使用资源有部分增加,但是编码速度达到了提高,而且能够进行错误检测,增加了其功能;同时由于 6B/5B 解码和 4B/3B 解码并行执行,减小了电路时序路径,工作频率达到工艺下的最高速度,提高了编码速度.

表 1 解码方案指标比较

技术指标 设计选择	使用资源	最高工作频率 /MHz
文献[4]	16LE	536.48
文献[5]	10LE	500.00
文献[7]	29LE	500
本设计	30LE	625

参考文献:

[1] Widmer A X, Franaszek P A. A DC-balance, partitioned-block, 8B/10B transmission code [J]. IBM

Journal of research and development, 1983, 23 (5):441.
[2] Chen H Y, Lin C H, Jou S J. Low-jitter transmission code for 4-PAM signaling in serial links[J]. IEEE Asia-Pacific Conference on Advanced System Integrated Circuits, 2004, 8(4):334-337.
[3] 许军,许西荣. PCI-Express 中 8b/10b 编解码器的设计与实现[J]. 微电子学与计算机, 2006, 23(3):37-39.
[4] 孙科达,石旭刚,史故臣. 一种新的 8B/10B 编解码方案设计与实现[J]. 光通信技术, 2009, 33(7):27-30.
[5] 武小强,田小平. 一种基于 FPGA 的 8B/10B 编解码电路的设计与实现[J]. 西安邮电学院报, 2010, 15(5):27-29.
[6] 刘智,宁红英,王普昌. 一种新的 8B/10B 编码电路设计[J]. 通信技术, 2009, 42(7):27-30.
[7] 刘文杰,施仨,郭林,等. 一种新的 8B/10B 编解码设计[J]. 光通信技术, 2012, 36(12):52-55.

作者简介:

邹陈女, (1990-), 硕士研究生. 研究方向为研究数字集成电路设计. E-mail: sandra@mail. edu. cn.
黄鲁男, (1961-), 副教授, 硕士生导师. 研究方向为混合信号 IC 芯片设计.
张步青男, (1981-), 博士. 研究方向为模拟与射频集成电路设计.

(上接第 27 页)

与传统的 SV 验证相比,本文利用 UVM 验证方法学优势产生随机化约束,并以 100%的覆盖率为目标不断完善测试用例,在可重用性方面既可以承用前人的 UVC,又为以后的 IP 验证工作提供了方便.

4 结束语

本文面向一款 MCU 中的 Radio 模块子系统,介绍了一种基于 UVM 验证方法学的混合信号验证环境,并成功应用于 Freescale(现 NXP)半导体 KW41 MCU 验证工作中.验证结果表明,该方法可以有效的发挥 UVM 验证方法学的在随机化约束、可重用性等方面的优势,可配置 Verilog-AMS 模型和模拟断言的引入使得验证更具可靠性和完备性,为混合信号的验证工作提供了一种思路和解决办法.

参考文献:

[1] Chang H, Kundert K. Verification of complex analog and RF IC designs [J]. Proceedings of the IEEE, 2007, 95(3):622-639.
[2] 杨华中,汪蕙,侯焕焕. 十亿晶体管级芯片系统集成所面

临的挑战与机遇[J]. 半导体情报, 2001, 38(1):2-4.
[3] Liang C, Zhong G, Huang S, et al. UVM-AMS based sub-system verification of wireless power receiver SoC [C]// Solid-State and Integrated Circuit Technology (ICSICT), 2014 12th IEEE International Conference on. China, Guilin, IEEE, 2014: 1-3.
[4] 田劲,王小力. 基于 UVM 验证方法学的 AES 模块级验证[J]. 微电子学与计算机, 2012, 29(8):86-90.
[5] Janick Bergeron. Writing testbench using system verilog[M]. Berlin: Synopsys, Inc, Springer, 2008.
[6] Srikanth Vijayaraghavan, Meyyappan Ramanathan. SystemVerilog Assertions 应用指南[M]. 陈俊杰,译. 北京:清华大学出版社, 2006.
[7] Accellera. Verilog-AMS language reference manual [M]. Napa, Accellera Systems Initiative Inc. 2014.

作者简介:

耿睿男, (1992-), 硕士研究生. 研究方向为混合信号设计验证. E-mail: mrgengrui@163. com.
王小力男, (1956-), 教授, 博士生导师. 研究方向为高性能 VLSI 与优化设计、集成化芯片系统设计与基础研究、光纤通讯与光波分复用(WDM)技术.